Automatic plotting of 5G parameters in NetSim

Software: NetSim Standard v13.0 (32-bit/ 64-bit), Visual Studio 2017/2019.

Project Download Link:

https://github.com/NetSim-TETCOS/NetSim_5G_LTE_Parameter_Log_and_Plots_v13_0_29/archive/refs/heads/main.zip

Follow the instructions specified in the following link to download and setup the Project in NetSim:

https://support.tetcos.com/en/support/solutions/articles/14000128666-downloading-and-setting-up-netsim-file-exchange-projects

Features

Using this workspace:

1. Users can plot Pathloss, Shadow Fading Loss, Total Loss, Rx_Power, SNR, Beam Forming Gain, MCS Index, and CQI Index vs. time using NetSim Plot.

2. Users can log Pathloss, Shadow Fading Loss, Total Loss, Rx_Power, SNR, Beam Forming Gain, MCS Index, and CQI Index with time stamps, to a CSV log file.

3. Users need to provide a file-based input (per a certain format) at the start of simulation for the parameters to be plotted or logged.

4. The plots are unique to
   a. Each gNB-UE pair
   b. Carrier ID
   c. DL or UL

5. The log entries are unique to
   a. Each gNB-UE pair
   b. Carrier ID
   c. DL or UL
   d. Each layer
6. The output parameters for different MIMO layers ($Min\ (N_t, N_r)$) are stacked in a single plot
7. Parameters are logged every slot time (1ms) and plotted.

8. There is no restriction in NetSim on the number of gNBs / UE in the network.

Example

In the below scenario

- The RAN portion has a MIMO layer count of 2, and both FastFading and ShadowFadingLoss are enabled.
- UE-10 moves in a straight line away from the gNB.
- The network is simulated for 60 s.

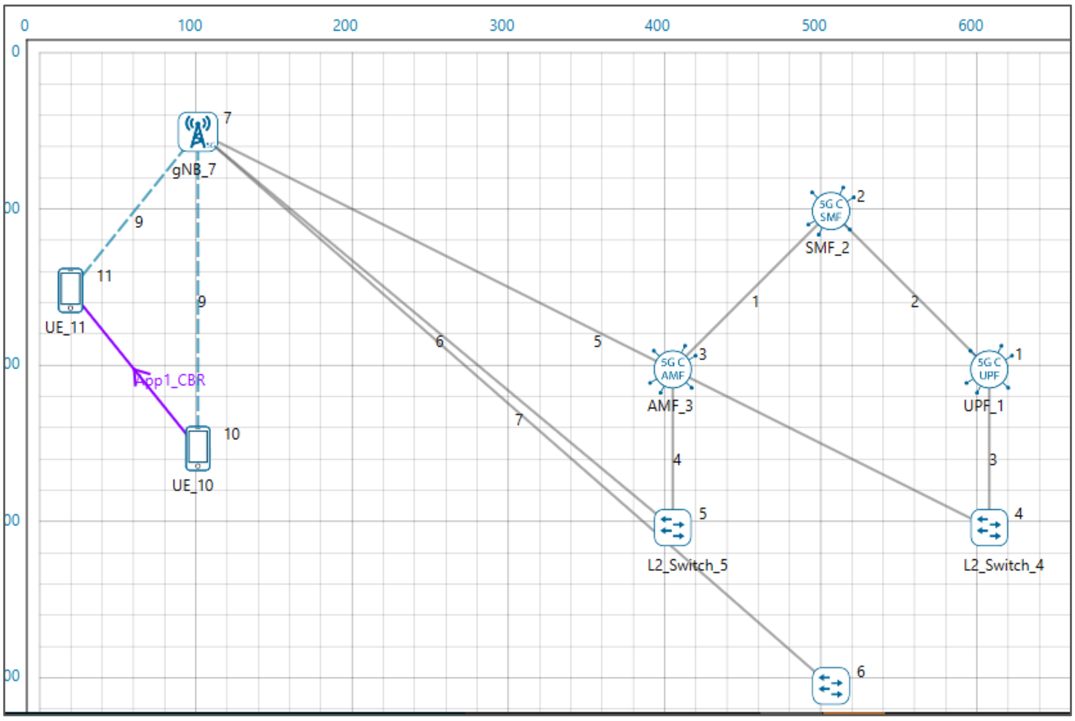


Figure 1: Network Topology in this experiment

- Upon running the simulation, a text file will open for the user to input the parameters and devices (tx-rx pair) for which parameters need to be logged and plotted. The input is per the format of <parameter>,<device1>,<device2> in the text file. To log gNB – UE (DL) data flow the gNB would be the 1st device while the UE would be the 2nd device. In the reverse direction (UL, UE to gNB) the UE would be the 1st device and the gNB the 2nd devices. In case of multiple gNBs, this input can be given for various gNB-UE pairs. Inputs are not case sensitive.

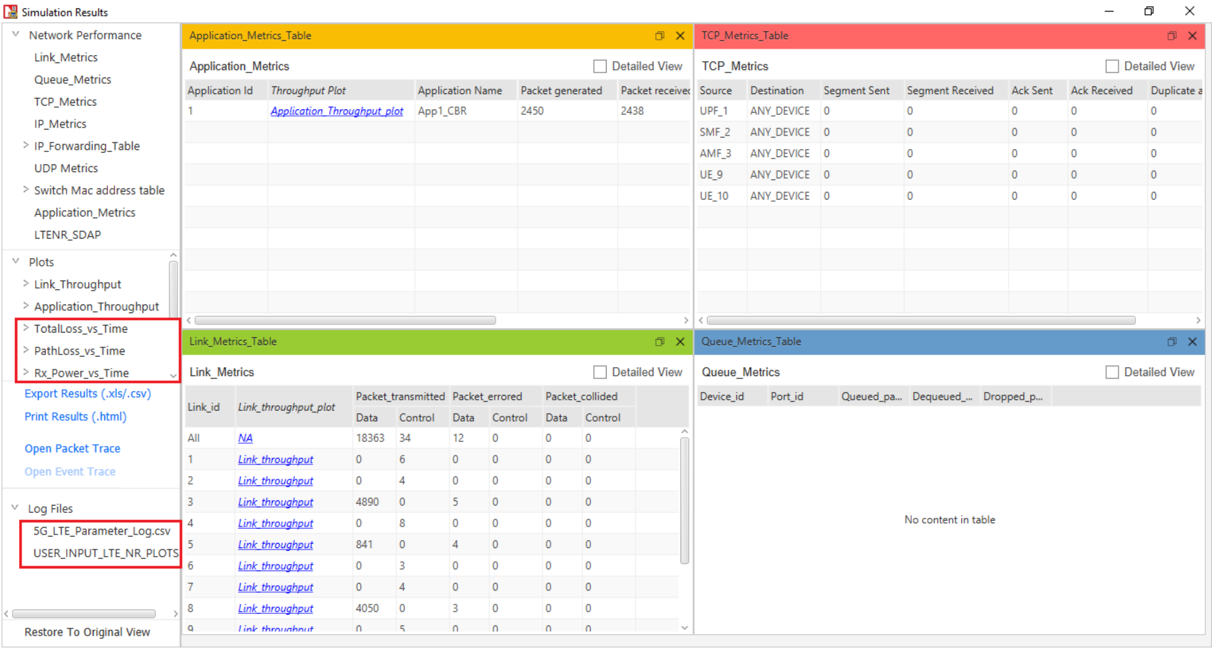
- For the above example, the input text file is as follows.

  ```
  TOTALLOSS,gNB_7,UE_10
  PATHLOSS,gNB_7,UE_10
  SHADOWFADINGLOSS, gNB_7,UE_11
  SHADOWFADINGLOSS,gNB_7,UE_10
  RX_POWER,gNB_7,UE_10
  SNR,gNB_7,UE_10
  BEAMFORMINGGAIN,gNB_7,UE_10
  CQI,gNB_7,UE_10
  MCS,gNB_7,UE_10
  SNR,gNB_7,UE_11
  ```

- Once the simulation starts, In the command prompt window it will show a message as "**Please update, Save and close the file and press any key to continue**".
- Add the parameters to be logged, close the input text file and press any key.
- Simulation starts running.

Results and discussion

Upon completion of simulation in the result window users can view the various plots.



Application_Metrics

Application Id	Throughput Plot	Application Name	Packet generated	Packet receivec
1	Application_Throughput_plot	App1_CBR	2450	2438

TCP_Metrics

Source	Destination	Segment Sent	Segment Received	Ack Sent	Ack Received	Duplicate a
UPF_1	ANY_DEVICE	0	0	0	0	0
SMF_2	ANY_DEVICE	0	0	0	0	0
AMF_3	ANY_DEVICE	0	0	0	0	0
UE_9	ANY_DEVICE	0	0	0	0	0
UE_10	ANY_DEVICE	0	0	0	0	0

Link_Metrics

Link_id	Link_throughput_plot	Packet_transmitted		Packet_errored		Packet_collided	
		Data	Control	Data	Control	Data	Control
All	NA	18363	34	12	0	0	0
1	Link_throughput	0	6	0	0	0	0
2	Link_throughput	0	4	0	0	0	0
3	Link_throughput	4890	0	5	0	0	0
4	Link_throughput	0	8	0	0	0	0
5	Link_throughput	841	0	4	0	0	0
6	Link_throughput	0	3	0	0	0	0
7	Link_throughput	0	4	0	0	0	0
8	Link_throughput	4050	0	3	0	0	0
9	Link_throughput	0	5	0	0	0	0

Queue_Metrics

Device_id	Port_id	Queued_pa...	Dequeued_...	Dropped_p...

No content in table

Figure 2: NetSim results dashboard with throughput highlighted

For each carrier, a separate plot is plotted with all the MIMO layers stacked in a single plot. The pathloss, shadow fading loss, and total loss remains same across the layers. Hence, for these parameters there is a single plot for all layers.

Result Plots

1. Pathloss Plot

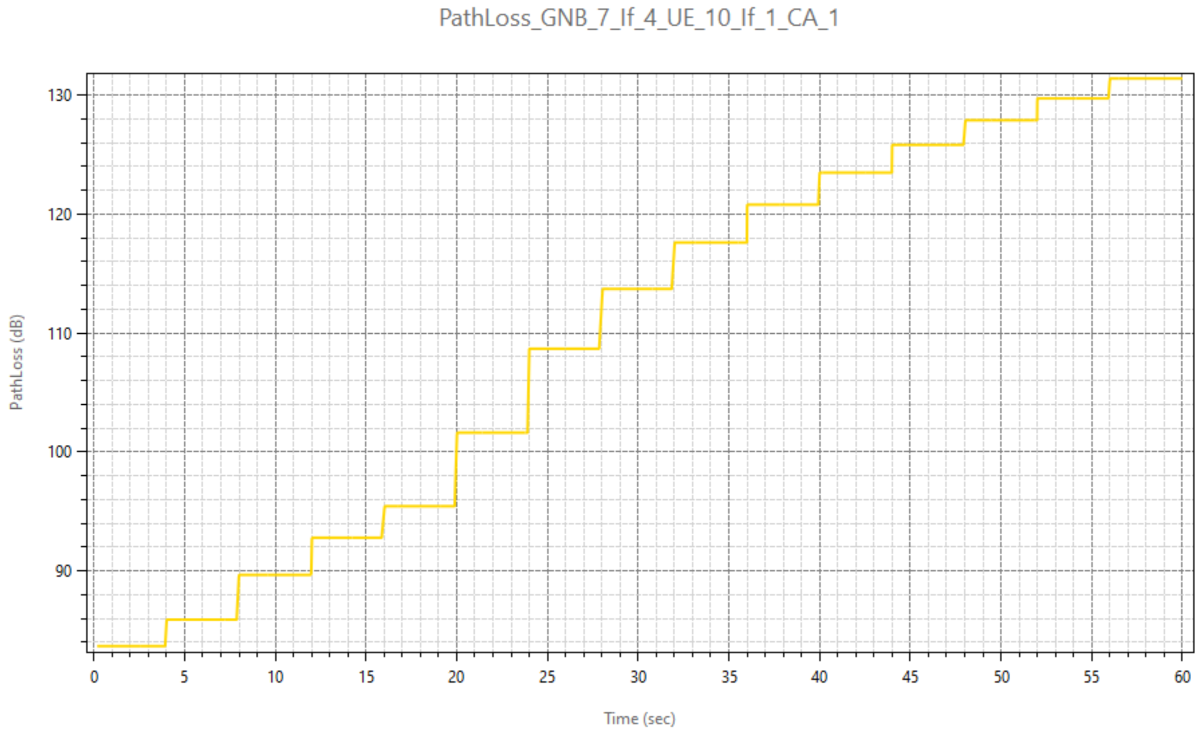


Figure 3: Pathloss Plot in NetSim

2. Total Loss (Shadow Fading loss plus Path loss)

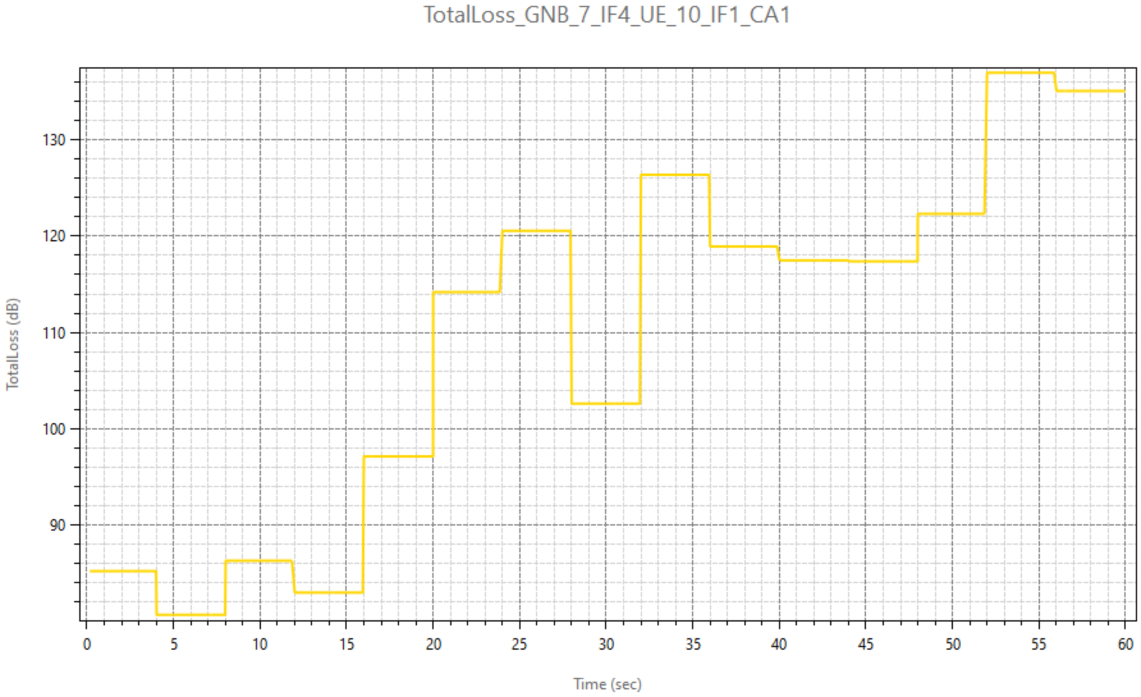


Figure 4: Total Loss (Shadow Fading loss plus Path loss) in NetSim

3. Shadow Fading Loss

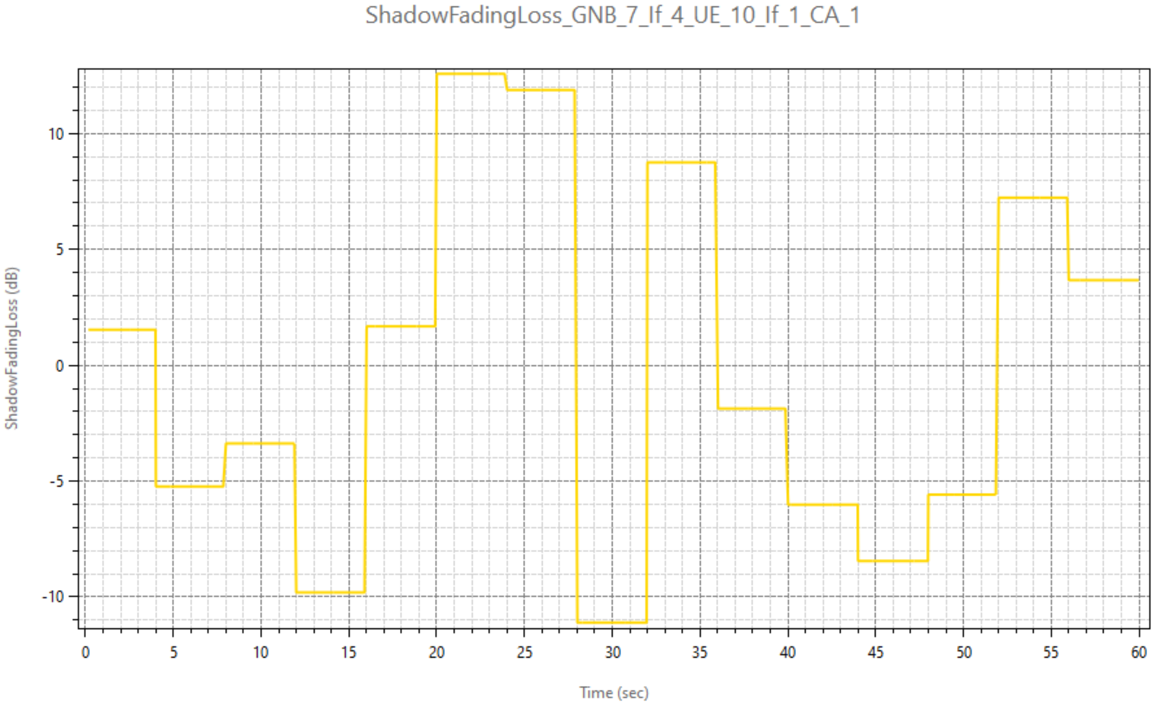


Figure 5: Shadow Fading Loss in NetSim

The plot title is ShadowFadingLoss_GNB_7_IF4_UE_10_IF1_CA1. And the naming convention is <ParameterType>_GNB_<ID>_IF<InterfaceID>_UE_<ID>_IF<InterfaceID>_CA<Carrier_ID>

4. Rx_Power Plot

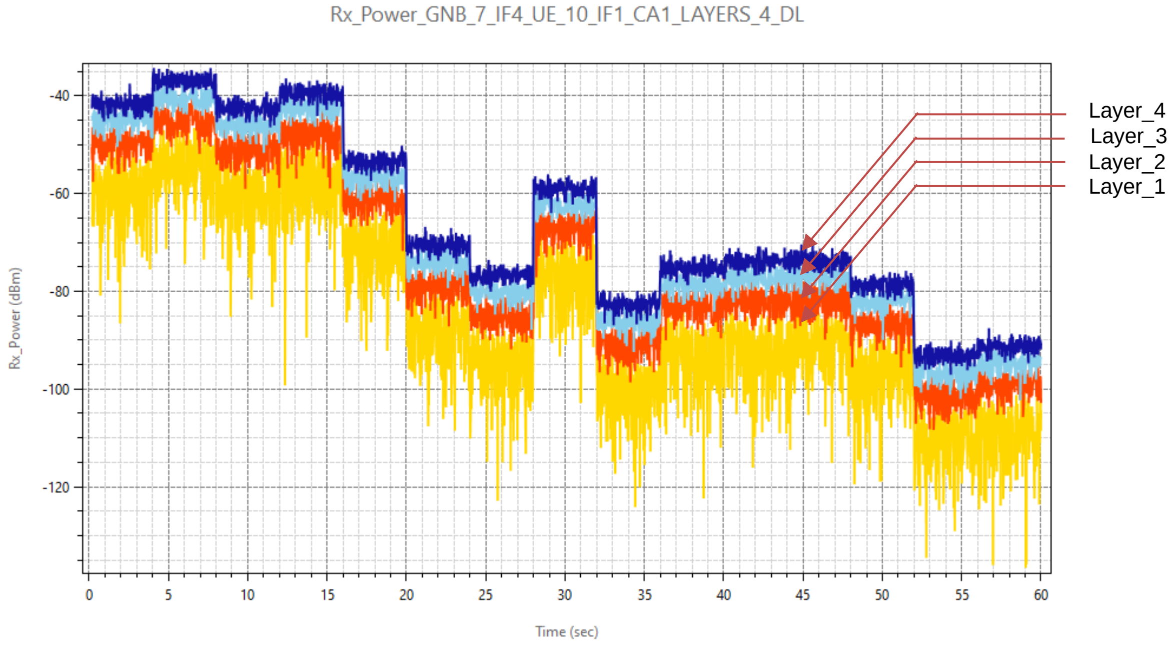


Figure 6: Rx_Power Plot in NetSim

5. SNR Plot

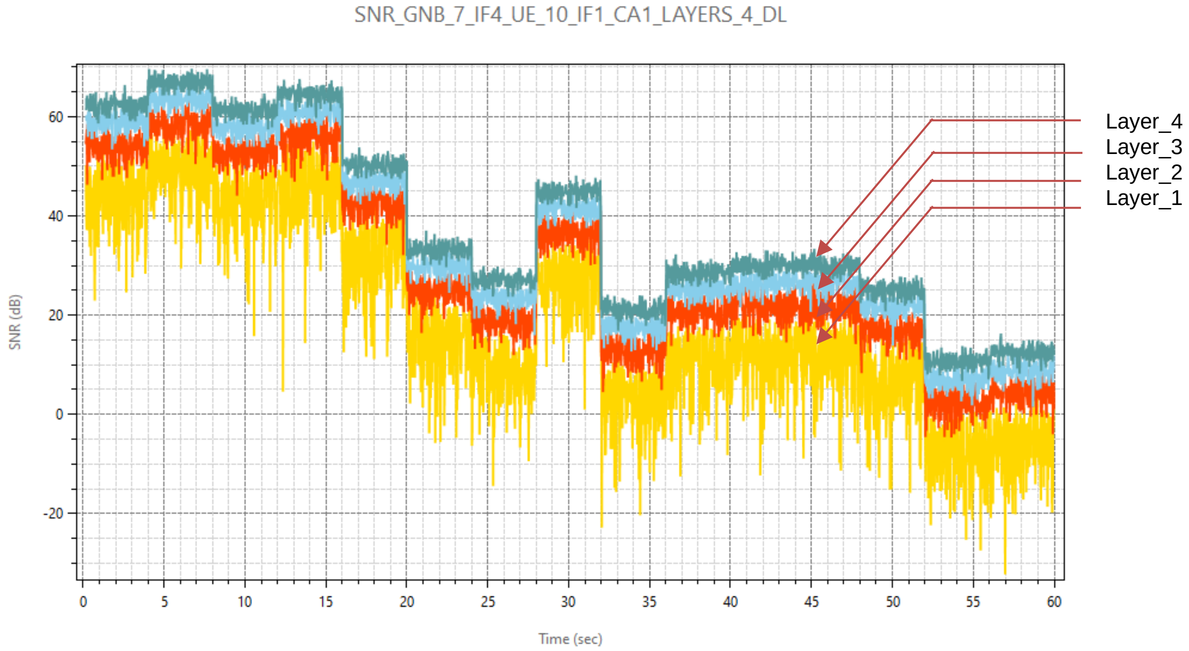


Figure 7: SNR Plot in NetSim

6. Beam Forming Gain

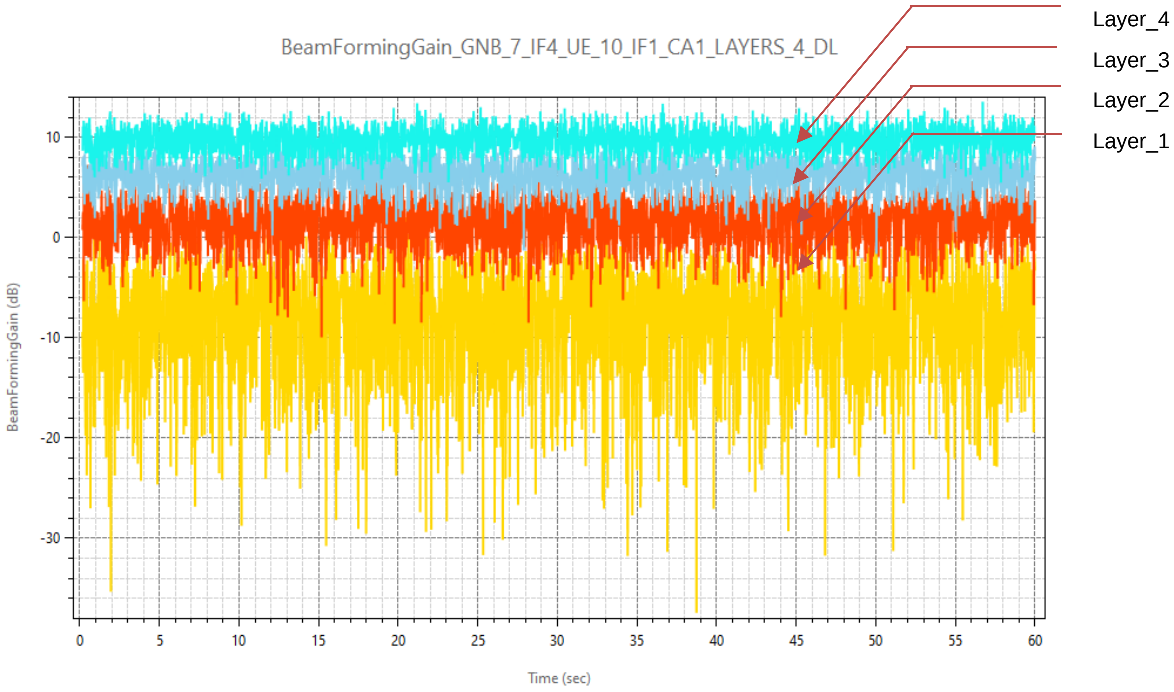


Figure 8: Beam Forming Gain in NetSim

7. CQI Index Plot

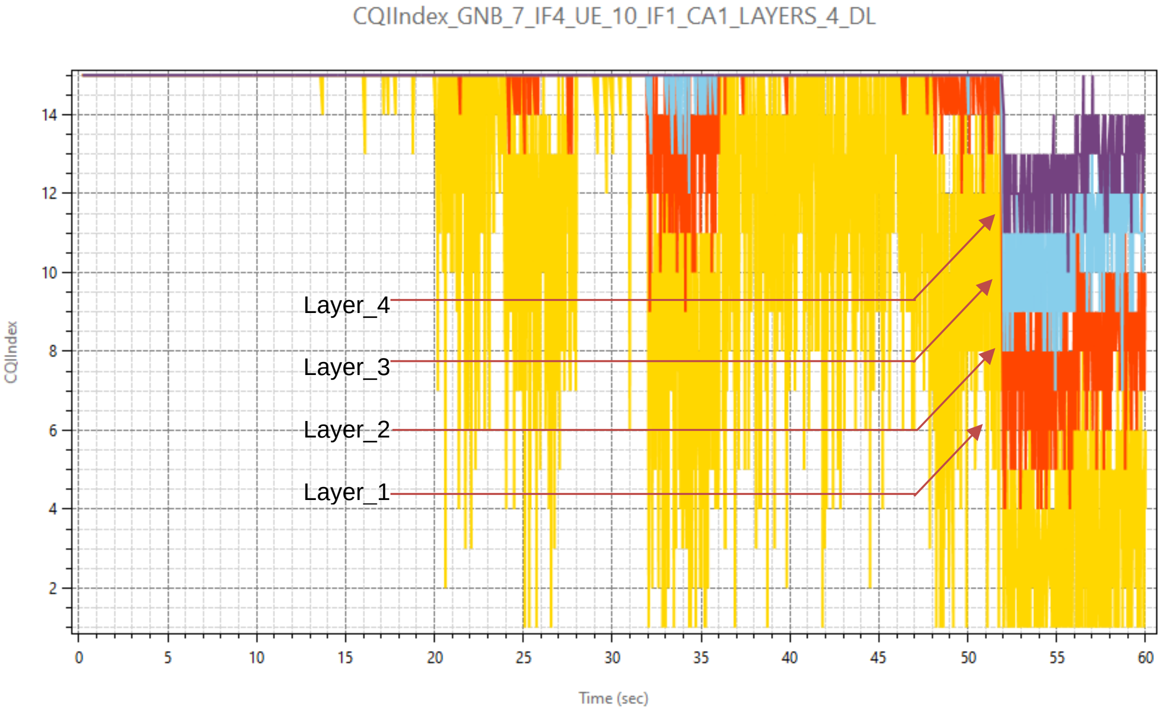


Figure 9: CQI Index Plot in NetSim

8. MCS Index Plot

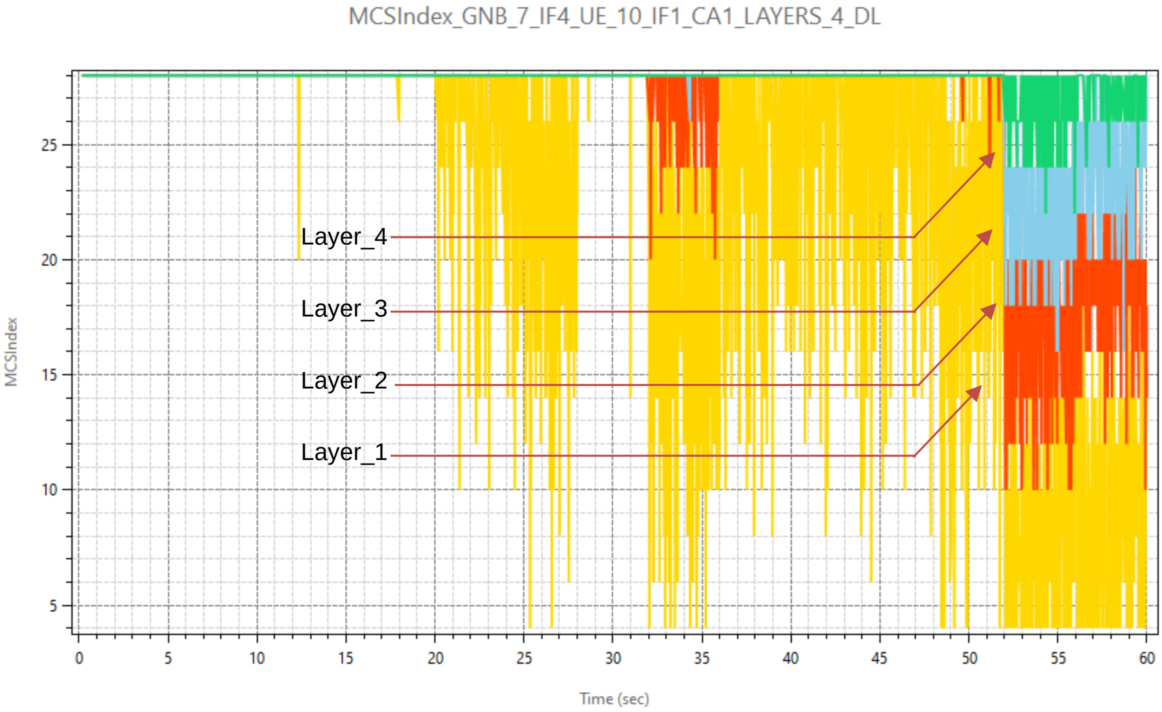
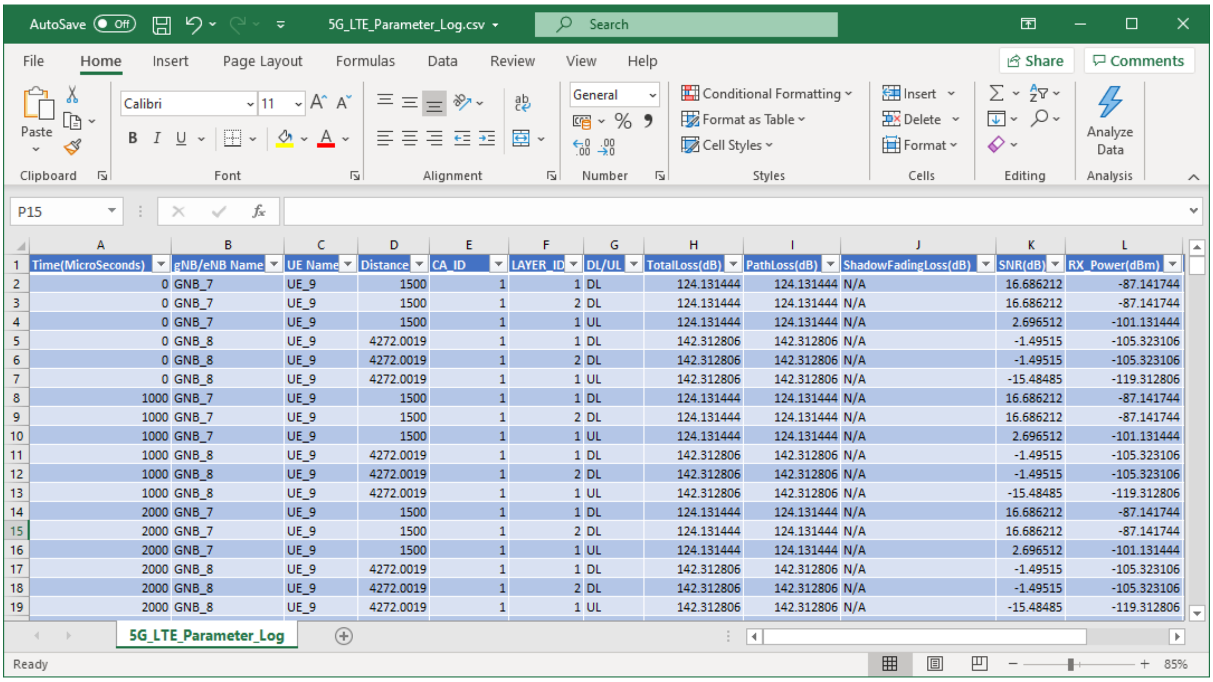


Figure 10: MCS Index Plot in NetSim

The SNR, Rx_Power, Beam Forming Gain, CQI Index, MCS Index plots are plotted for all MIMO layers for a Carrier 1. In the chart title layer count and application direction (DL/UL) are also present.

Parameter log file



	A	B	C	D	E	F	G	H	I	J	K	L
1	Time(MicroSeconds)	gNB/eNB Name	UE Name	Distance	CA_ID	LAYER_ID	DL/UL	TotalLoss(dB)	PathLoss(dB)	ShadowFadingLoss(dB)	SNR(dB)	RX_Power(dBm)
2	0	GNB_7	UE_9	1500	1	1	DL	124.131444	124.131444	N/A	16.686212	-87.141744
3	0	GNB_7	UE_9	1500	1	2	DL	124.131444	124.131444	N/A	16.686212	-87.141744
4	0	GNB_7	UE_9	1500	1	1	UL	124.131444	124.131444	N/A	2.696512	-101.131444
5	0	GNB_8	UE_9	4272.0019	1	1	DL	142.312806	142.312806	N/A	-1.49515	-105.323106
6	0	GNB_8	UE_9	4272.0019	1	2	DL	142.312806	142.312806	N/A	-1.49515	-105.323106
7	0	GNB_8	UE_9	4272.0019	1	1	UL	142.312806	142.312806	N/A	-15.48485	-119.312806
8	1000	GNB_7	UE_9	1500	1	1	DL	124.131444	124.131444	N/A	16.686212	-87.141744
9	1000	GNB_7	UE_9	1500	1	2	DL	124.131444	124.131444	N/A	16.686212	-87.141744
10	1000	GNB_7	UE_9	1500	1	1	UL	124.131444	124.131444	N/A	2.696512	-101.131444
11	1000	GNB_8	UE_9	4272.0019	1	1	DL	142.312806	142.312806	N/A	-1.49515	-105.323106
12	1000	GNB_8	UE_9	4272.0019	1	2	DL	142.312806	142.312806	N/A	-1.49515	-105.323106
13	1000	GNB_8	UE_9	4272.0019	1	1	UL	142.312806	142.312806	N/A	-15.48485	-119.312806
14	2000	GNB_7	UE_9	1500	1	1	DL	124.131444	124.131444	N/A	16.686212	-87.141744
15	2000	GNB_7	UE_9	1500	1	2	DL	124.131444	124.131444	N/A	16.686212	-87.141744
16	2000	GNB_7	UE_9	1500	1	1	UL	124.131444	124.131444	N/A	2.696512	-101.131444
17	2000	GNB_8	UE_9	4272.0019	1	1	DL	142.312806	142.312806	N/A	-1.49515	-105.323106
18	2000	GNB_8	UE_9	4272.0019	1	2	DL	142.312806	142.312806	N/A	-1.49515	-105.323106
19	2000	GNB_8	UE_9	4272.0019	1	1	UL	142.312806	142.312806	N/A	-15.48485	-119.312806

Figure 11: 5G Log file parameter

The 5G_LTE_Parameter_Log.csv file logs the details of parameters specified in the input file with respect to time.

Appendix: NetSim source code modifications

Open the Source codes in Visual Studio by going to Your work-> Workspace Options and Clicking on Open code button.

To the in LTE_NR project, files LTE_NR_Plot.c and LTE_NR_Parameter_Log.c has been added. These files contain the definitions of the functions that responsible for plotting and logging parameters associated with 5G/LTE networks in NetSim.

The function fn_NetSim_LTE_NR_Init_Plots and fn_NetSim_LTE_NR_init_Parameter_Log has been called in LTENR.c file for initializing the plot.

```
static bool isplotinit= false;
//Function prototype
int fn_NetSim_LTE_NR_Init_F();
int fn_NetSim_LTE_NR_Configure_F(void** var);
int fn_NetSim_LTE_NR_Finish_F();
#pragma endregion

#pragma region LTENR_INIT
_declspec(dllexport) int fn_NetSim_LTE_NR_Init()
{
   if(!isplotinit)
   {
   fn_NetSim_LTE_NR_Init_Plots();
   fn_NetSim_LTE_NR_init_Parameter_Log();
   isplotinit= true;
   }
return fn_NetSim_LTE_NR_Init_F();
}
```

The initialization of functions and functions to update the logs for plotting and logging to CSV file has been made as follows in LTENR_handleStartSlotEvent function.

```
void LTENR_handleStartSlotEvent()
{
NETSIM_ID gnbId = pstruEventDetails->nDeviceId;
NETSIM_ID gnbIf = pstruEventDetails->nInterfaceId;
ptrLTENR_GNBPHY phy = LTENR_GNBPHY_GET(gnbId, gnbIf);

#pragma warning (disable : 4047)
int CA_ID = pstruEventDetails->szOtherDetails;
#pragma warning (default : 4047)

ptrLTENR_CA ca = phy->spectrumConfig->CA[CA_ID];

LTENR_resetSlot(phy, CA_ID);
print_ltenr_log("Starting new slot for gNB %d:%d\n", gnbId, gnbIf);
print_ltenr_log("CA_ID for Slot = %d\n", CA_ID);
print_ltenr_log("\tFrame Id = %d\n", phy->frameInfo[CA_ID]->frameId);
print_ltenr_log("\tSubFrame Id = %d\n", phy->frameInfo[CA_ID]->subFrameId);
print_ltenr_log("\tSlot Id = %d\n", phy->frameInfo[CA_ID]->slotId);
print_ltenr_log("\tSlot start time (us) = %lf\n", phy->frameInfo[CA_ID]->slotStartTime);
print_ltenr_log("\tslot end time (us) = %lf\n", phy->frameInfo[CA_ID]->slotEndTime);
print_ltenr_log("\tSlot type = %s\n", strLTENR_SLOTTYPE[phy->frameInfo[CA_ID]->slotType]);
phy->currentFrameInfo = phy->frameInfo[CA_ID];
phy->currentFrameInfo->Current_CA_ID = CA_ID;
if (phy->frameInfo[CA_ID]->slotId != ca->slotPerSubframe)
        LTENR_addStartSlotEvent(gnbId, gnbIf,
                phy->frameInfo[CA_ID]->slotEndTime, CA_ID);

ptrLTENR_ASSOCIATEDUEPHYINFO info = phy->associatedUEPhyInfo;
//
if (pstruEventDetails->dEventTime ==0 || pstruEventDetails->dEventTime==200000)
{
        for (NETSIM_ID d = 1; d <= NETWORK->nDeviceCount; d++)
        {
                for (NETSIM_ID in = 1; in <= DEVICE(d)->nNumOfInterface; in++)
                {
                        if (!isLTE_NRInterface(d, in))
                                continue;
                        if (!isGNB(d, in))
                                continue;

                        ptrLTENR_GNBPHY phy_ = LTENR_GNBPHY_GET(d, in);
                        ptrLTENR_ASSOCIATEDUEPHYINFO info_ = phy_->associatedUEPhyInfo;

                        while (info_)
                        {
                                fn_NetSim_LTE_NR_init_PropagationInfo_Plots(phy_, info_);
                                fn_NetSim_LTE_NR_init_Power_Plots(phy_, info_);
                           LTENR_ASSOCIATEDUEPHYINFO_NEXT(info_);
                        }

                }
        }
}
while (info)
{
        if (info->isAssociated)
        {
                for (NETSIM_ID i = 0; i < phy->ca_count; i++)
                        LTENR_PHY_setAMCInfo(phy, info, i);
        }

        fn_NetSim_LTE_NR_add_PropagationInfo_Plot_data(info, CA_ID);
        fn_NetSim_LTE_NR_add_Power_Plot_data(info, CA_ID);
        ptrINFO param_info = parameter_log_info;
        if (param_info->isParameterlog)
                fn_NetSim_LTE_NR_Log_Parameters(phy, CA_ID, info);
        info = LTENR_ASSOCIATEDUEPHYINFO_NEXT(info);
}

LTENR_NotifyMACForStartingSlot();

phy->frameInfo[CA_ID]->prevSlotType = phy->frameInfo[CA_ID]->slotType;
}
```

For adding plot data at every slot time (1 ms) the below highlighted function has been used in LTENR_phy.c file.

```
static void LTENR_PHY_setAMCInfo(ptrLTENR_GNBPHY phy,
ptrLTENR_ASSOCIATEDUEPHYINFO info, int CA_ID)
{
UINT layerCount;
ptrLTENR_UEPHY uePhy = LTENR_UEPHY_GET(info->ueId, info->ueIf);

//Downlink
layerCount = LTENR_PHY_GET_DLLAYER_COUNT(uePhy);
for (UINT i = 0; i < layerCount; i++)
{
        print_ltenr_log("\tAMC info between gNB %d:%d and UE %d:%d, Carrier Id = %d, Layer Id =
%d for downlink-\n",
                phy->gnbId, phy->gnbIf,
                info->ueId, info->ueIf,
                CA_ID, i);
        info->downlinkAMCInfo[CA_ID][i]->SpectralEfficiency =
LTENR_PHY_GetDownlinkSpectralEfficiency(info->propagationInfo[CA_ID], i);
        setAMCInfo(phy, info->downlinkAMCInfo[CA_ID][i]);
}

//Uplink
layerCount = LTENR_PHY_GET_ULLAYER_COUNT(uePhy);
for (UINT i = 0; i < layerCount; i++)
{
        print_ltenr_log("\tAMC info between gNB %d:%d and UE %d:%d, Carrier Id = %d, Layer Id =
%d for uplink-\n",
                phy->gnbId, phy->gnbIf,
                info->ueId, info->ueIf,
                CA_ID, i);
        info->uplinkAMCInfo[CA_ID][i]->SpectralEfficiency =
LTENR_PHY_GetUplinkSpectralEfficiency(info->propagationInfo[CA_ID], i);
        setAMCInfo(phy, info->uplinkAMCInfo[CA_ID][i]);
}
fn_NetSim_LTE_NR_add_AMCInfo_Plot_data(info, CA_ID);
}
```

Disabling Plotting/Logging

Generation of plots or the parameter log can be disabled by commenting the function calls in the fn_NetSim_LTE_NR_Init() function. The function call fn_NetSim_LTE_NR_Init_Plots can be commented to disable plots and the function call fn_NetSim_LTE_NR_init_Parameter_Log can be commented to disable generation of a parameter log CSV file.

```
_declspec(dllexport) int fn_NetSim_LTE_NR_Init()
{
   if(!isplotinit)
   {
   fn_NetSim_LTE_NR_Init_Plots(); //comment line to disable plots
   fn_NetSim_LTE_NR_init_Parameter_Log(); //comment line to disable parameter log
   isplotinit = true;
   }
return fn_NetSim_LTE_NR_Init_F();
}
```